基于区域划分的混合式 NDN 缓存策略

宋彩凤[1,2]，李　翀[1†]，刘学敏[1]

(1. 中国科学院计算机网络信息中心，北京 100190; 2. 中国科学院大学，北京 100049)

摘　要：针对 NDN(命名数据网络)中确定性缓存和概率性缓存进行研究，提出一种确定性缓存和概率性缓存相结合的混合式 NDN 缓存策略(HDP)。基于区域划分的思想，在网络边缘采用基于热度的确定性缓存策略，在网络核心采用基于缓存收益和内容热度的概率性缓存策略，从而将两种缓存策略的优势相结合，进一步提高 NDN 缓存性能。仿真实验表明该策略与现有 NDN 缓存方法相比，能有效提高缓存服务率和命中率，并有助于降低内容访问延迟，改善用户体验。

关键词：命名数据网络；缓存技术；确定性缓存；概率性缓存；混合式缓存

中图分类号：TP393　　doi: 10.3969/j.issn.1001-3695.2018.02.0051

Hybrid NDN caching strategy based on region division

Song Caifeng[1, 2], Li Chong[†1], Liu Xuemin[1]

(1. *Computer Network Information Center*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*; 2. *University of Chinese Academy of Sciences*; *Beijing* 100190, *China*)

Abstract: Considering the advantages and disadvantages of both deterministic caching strategies and probabilistic caching strategies in Named-Data Networking, this paper proposed a hybrid caching strategy (named HDP) . Based on the idea of region division, the hybrid NDN caching strategy used the popularity-based deterministic caching strategy at the edge of the network, and used the probabilistic caching strategy based on cache benefit and content popularity at the core area. Combining the advantages of both cache strategies, the performance of NDN caching is enhanced. Simulation results show that compared with current NDN methods this strategy is able to increase cache service ratio and hit ratio and reduce content access delay to improve the user experience.

Key words: Named-Data Networking; caching technology; deterministic caching; probabilistic caching; hybrid caching

0　引言

当前基于 TCP/IP 的互联网采用基于位置信息的传输模式，用户请求内容需要事先知道内容源的位置。然而随着互联网的发展，当前用户更多地关注内容本身而并不太关注内容来源的位置，这造成底层网络传输模式与上层用户应用需求间存在矛盾，不利于互联网的发展。针对上述问题，命名数据网络（named data networking,NDN）[1, 2]应运而生，成为一种具有重要影响力的未来网络发展方向。NDN 网络以内容为中心，主要面向内容获取与分发类应用。

网内缓存是 NDN 的重要特征，对于改善内容分发效率具有重要作用。NDN 路由器均可对网络中途经内容进行缓存，当内容被缓存后可为后续请求提供服务，从而达到降低内容响应延迟，减少网络流量，缓解服务器负载等目的。

当前 NDN 缓存方案可大致分为确定性缓存[1~7]和概率性缓存[8-11]两大类。在确定性缓存方案中，网络中路由器采用确定性的方式决策如何对途经内容进行缓存，缓存决策的优劣取决于所搜集的决策信息的多少，因此如果想要做出较优的决策，则会导致较高的信息搜集和通信开销。概率性缓存主张每个路由器按照一定概率进行缓存，一定程度上降低了缓存内容间的冗余度，但难以确定每次缓存后的结果，系统稳定性较差，收敛性不佳。

针对确定性缓存和概率性缓存的各自特点，本文提出一种混合式的内容缓存方法(Hybrid strategy combining Deterministic caching with Probabilistic caching, 简称为HDP)，将两者优势相结合，进一步优化NDN缓存性能。其主要思想是在网络边缘采用确定性缓存方案，借助最优化决策方法优化缓存内容的放置；同时在网络核心采用基于概率的缓存方案，降低缓存冗余度，提高缓存空间利用率。

本文的主要贡献如下：

收稿日期：2018-02-13；修回日期：2018-03-26

作者简介：宋彩凤（1991-），女，山东聊城人，硕士研究生，主要研究方向为未来网络关键技术、网内缓存（songcaifeng@cnic.cn）；李翀（1978-），男（通信作者），安徽霍邱人，副研究员，博士，主要研究方向为网络协同工作环境、分布式计算、软件体系结构等；刘学敏（1975-），男，山东烟台人，高级工程师，硕士，主要研究方向为网络协同技术、云计算、容器和微服务技术.

a）在分析现有确定性缓存和概率性缓存方案各自优劣的基础上，提出一种确定性和概率性缓存相结合的基于区域划分的缓存架构，结合不同区域特点，将两种缓存方法有机地结合。

b）在网络核心区域提出一种基于概率的缓存算法，能有效适应核心区域特点；在网络边缘区域采用基于最优化理论的缓存算法，从而进一步优化NDN缓存性能。

c）定义NDN缓存相关性能评价指标，在多种场景下开展大量仿真实验，并与现有NDN确定性缓存和概率性缓存方案进行对比。实验结果表明，HDP方案相比于对比方案在缓存服务命中率、缓存命中率、响应延迟等方面均具有显著性能优势。

1 相关研究

1.1 确定性缓存

确定性缓存，是指当内容途经 NDN 网络中路由器时，路由器以某种方式确定性的存或不存该数据内容。缓存效果比较依赖于已知信息的多少，越多决策信息已知时（如用户请求情况、网络拓扑、缓存容量等信息），该类方案越容易获得比较理想的缓存效果。但当决策信息较少时，该类方法难以提高缓存效率，有时会导致缓存冗余的产生，系统缓存服务率较低。代表性的缓存方案如下：

NDN 最初采用处处缓存的方法（leave cache everywhere, LCE）[1-2]，数据消息在 NDN 网络传输过程中，途径的所有路由器节点均会将其缓存。显而易见，该方案会造成在传输路径上存在大量内容重复存储的现象，缓存空间得不到充分利用，缓存命中率有待提高。

LCD 和 MCD[3, 4]是针对 Web 缓存提出的缓存方案。LCD 方案是一种相比于 LCE 降低缓存冗余度的方法，主要思路是：当请求消息在某缓存节点发生内容命中时，当数据回传时，当前节点的下一跳节点会缓存此内容，其他下游节点并不缓存。这种方法会呈现出一种热门内容逐渐向用户端移动的趋势。MCD 是对 LCD 的进一步改进，当用户请求消息命中缓存时，在下游节点缓存内容的同时，当前节点也会删除该内容，进一步降低缓存的冗余度。

ABC[5]是围绕内容存在时间设计的缓存策略。内容块经过路由器时会被缓存，同时每个内容被分配一个存在时间（age）。内容在缓存节点中停留的时间由存在时间决定，即每次替换超时的内容。存在时间与内容的热度和缓存的位置密切相关。内容越热门，缓存位置越靠近用户，内容存在时间越长。通过合理地替换内容间接达到了内容放置的目的。

CINC[6]是一种协作式的确定性缓存机制。相互协作的所有缓存节点会事先分配唯一的编号。当某个缓存节点收到新到内容块时，会根据内容块的编号结合哈希算法，计算存放该内容块的节点编号；在收到用户请求时，按照同样哈希计算方法，可以将请求路由到相应的缓存节点。节点编号的设计方法对性能有重要影响，需合理设计。

Wang 等人[7]提出一种在已知用户请求分布、缓存系统总容量、网络拓扑等信息已知的情况下的最优缓存放置方法。借助建立最优化模型并求解，可获得在系统缓存总容量一定的情况下，系统获得最小延迟时的内容放置情况。这说明当系统已知信息足够多时，可获得最优的确定性缓存算法。

1.2 概率性缓存

NDN 路由以一定概率确定是否缓存经过的内容。该类方法的优点是可以在一定程度上缓解缓存方法内容冗余度高、替换操作过多等问题。同时概率性缓存也存在最终缓存内容分布具有不确定性，系统收敛性和稳定性较弱等缺点。代表性的缓存方案有以下几种：

RAC[8]是一种针对 LCE 的改进的概率性方法。从缓存命中节点到请求者之间的所有缓存节点均按照统一的概率缓存该内容。当缓存概率为 1 时，该方法则演变为 LCE。这是一种按照固定概率在全网进行缓存的方案。基于固定概率的缓存，一定程度上降低缓存内容冗余度，然而概率值的大小难以确定，需要借助长期网络运维经验获得。

ProbCache[9, 10]是一种基于概率的缓存机制，在缓存内容时同时兼顾缓存路径的剩余容量以及缓存收益：当前节点下游路径中所有节点的缓存容量之和越大，则当前节点越可能缓存内容。缓存节点离请求者越近，表明收益越大，则缓存的概率也越大。然而该方法没有考虑内容热度间的差异。

CRCache[11]是另一种基于概率的缓存算法。将内容按照热度高低分成若干级别，级别越高内容越热。按照路由器所在的位置（重要性）分为多个级别，级别越高路由器的重要性越大。具体而言，当内容在经过路由器时，计算内容重要性和路由器重要性的差值，如果差值越小，则内容在此处缓存的概率就越高。因此，此种方法的目标是将内容按照级别放在与级别与之相近的路由器中，从而达到重要位置的路由器存放重要信息，次要位置的路由器存放次要信息的目的。

2 NDN 基本原理

NDN（named data networking）是未来网络重要发展方向。NDN 网络的设计针对未来以内容分发为主的应用需求，研究以内容为中心的体系结构，力争解决传统 TCP/IP 网络在扩展性、安全性、移动性等方面的问题。

NDN 主要具有以下特性：a）网内缓存,在路由器中缓存内容，从而提高内容分发效率;b）内容安全,在 NDN 中每个内容具有唯一标志的名字，名字和内容存在一一对应关系，同时内容名字和内容借助签名机制，保证内容本身的安全性；c）按名路由[12-14],请求消息的路由是在 NDN 路由器中按照名字逐跳转发的,请求消息的路径和数据返回的路径是同一路径,除外借助在网络中保持路由状态信息，可以实现内容的多路径（multicast）传输，提高内容分发效率;d）移动性支持较好,由于用户移动位置后，仅需向网络重新发送请求即可获取内容（不需要三角路由），从而可大大改善用户移动性，特别是请求者的移动性。

NDN 网络主要存在请求消息（interest message）和数据消

息（data message）两类消息。请求消息中包含内容名字，NDN 路由器基于内容名字对请求消息进行逐跳转发，直到命中缓存节点或服务器。此后数据沿着同一路径反向传回到请求者。

NDN 路由器维护 CS 表(content store)、PIT 表(pending interest table)和 FIB 表(forwarding information base)。CS 表用于缓存内容。请求消息的传输路径记录在 PIT 表中，协助数据按原路径反向传输。FIB 表借助 NDN 节点相互间通信形成，请求消息结合 FIB 内容进行逐跳转发。

当 NDN 路由器接收到请求消息后，会按照 CS 表、PIT 表和 FIB 表的顺序查找。如果 CS 表命中则直接将数据返回给用户，否则再查 PIT 表。如果 PIT 表此前没记录，则将该请求记录在表中。如果 PIT 表存在相同请求的记录，则将请求合并到请求列表中。最后查 FIB 表，并进行相应转发。当 NDN 路由器收到数据消息时，根据 PIT 表中记录的反向路径传输。

3 HDP 方案设计

3.1 HDP 方案概述

HDP 方案主要分为三部分:a）核心区域和边缘区域的划分方法；b）在网络边缘采用确定性算法,由于离用户较近，便于感知用户行为,网络范围相对较小,系统便于统计相关决策信息，因此借助最有化方法的确定性算法可以获得较好的解决方案；c)在网络核心采用概率性算法。由于距离用户较远，范围较大，很多网络信息难以获得,因此适合于采用概率性方法进行缓存。

3.2 网络边缘与核心的划分方法

关于网络边缘与核心的划分方法，本节将从以下几方面进行阐述。首先，核心区域和边缘区域划分方法，本文充分参考现有的相关研究成果，采用一种启发式基于节点度的方法。例如给节点度数设置一个阈值 α，所有节点度数小于等于 α 的节点构成边缘网络，所有节点度数大于 α 的节点构成核心网络。此种方法的准确性和合理性已在一些论文中通过验证，如参考文献[18]。其次，本文的主要贡献在于提出按照边缘和核心的不同区域，分别采用确定性和概率性缓存不同策略，而不是像传统方法在全网内采用单一缓存策略。本文所提的区域划分方法是一种初步设计方法，接下来我们也计划设计更为精细的方案。最后，区域划分作为一个开放性的问题，本文也同时给出了几种其他可能的候选方案，供用户结合实际需求进行合理选择。比如网络服务提供商 ISP 可以根据自身网络层次结构，合理划分边缘和核心网络。再如，对于自身有划分的网络，如运行 OSPF 的域内网络，OSPF 骨干网可作为网络核心，每个 OSPF 非骨干区域可作为网络边缘。

3.3 核心区域的缓存算法

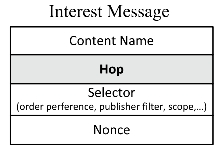
在核心区域内每个NDN路由器（缓存节点）对所有经过的数据块，按照一定概率决策是否对其缓存，具体缓存公式如下所示：

$$p = \frac{N - r_n + 1}{N} \times \frac{d}{D_{core}} \qquad (1)$$

其中:N表示网络中内容块的数量，r_n代表内容热度排名，d表示数据消息（data message）经过的跳数。数据消息中有字段记录该跳数值。初始为0，每到达一个路由器其值加1。D_{core}表示数据消息返回时在核心区域内所能经过的最大路径长度（跳数）。

从式（1）可以看出，乘积两侧分别表示内容热度和缓存收益。因此，核心区域内缓存策略的主要思想是内容热度越高、缓存收益越大的内容块，其被缓存的概率就越高。

为方便统计式（1）中的相关信息，对NDN相关消息格式、CS表和FIB表进行细微调整。如下图1所示，对NDN请求消息和数据消息格式进行扩展，分别增加了Hop字段和Hop和Distance字段，如下图所示。当请求消息到达核心区域的第一个路由器时，Hop字段初始值为0，每到达一个路由节点该值加1。当请求命中缓存节点或服务器时，该值被记录在数据消息Distance字段中，并被返回。



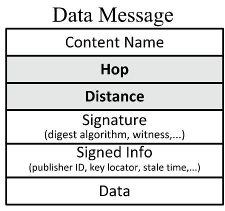


图 1 请求消息格式

与此同时也在CS表中增加Hop字段，在FIB条目中增加Hop字段，以分别应对命中缓存时和发生多次请求消息聚合时的相关信息统计。

服务器相应操作

核心区域内的服务器收到Interest消息后，需要准备相应的Data消息。一方面，将Interest消息的Hop赋值给Data消息中Distance字段，另一方面，将Data消息中Hop字段初始化为1，并返回Data消息。

路由器算法

路由器中的算法可分为对Interest消息的处理算法以及对Data消息的处理算法两部分，具体内容参见下表。如表1所示，核心区域路由器对Interest消息的处理仍是依次查CS表、PIT表和FIB表。如表2所示，核心区域路由器对Data消息的处理方法主要是，以式（1）计算的概率缓存经过的内容，并结合PIT记录信息转发Data消息。

值得注意的是，式（1）中N和r_n均可通过路由器统计获得，因此主要工作在于如何统计d和D_{core}。

数据消息返回时可分为两种情况：

a)请求消息命中服务器后返回，此时Interest消息中Hop字段记录了D_{core}的值，Data返回时该值携带在其Distance字段中。

（参见服务器相关操作）。

b)请求消息命中缓存后返回，此时Interest消息中的Hop字段需要加上缓存节点与服务器间的距离才是D_{core}。（参见表1中第5行。值得注意的是，此前需要在内容缓存时同时将缓存节点与服务器间距离记录在缓存的字段中，参见表2第3行）

表 1 核心区域路由器对 Interest 消息的处理算法

核心区域路由器对 Interest 消息的处理算法

```
1.  查 CS 表;
2.   if（缓存命中）
3.      对 Data 消息中字段做如下改动:
4.      Hop = CS 中 Hop 字段值;
5.      Distance=请求消息中 Hop 值 + CS 表中 Hop 值;
6.      返回 Data 消息;
7.   else   /* 缓存不命中时查 PIT 表 */
8.      if（没有匹配的 PIT 表项）
9.         创建一个新的 PIT 表项;
10.     else（有匹配的 PIT 表项）
11.        将请求消息到达接口添加到相应接口列表中;
12.        相应接口的 Hop = 请求消息中的 Hop;
13.     end if
14. 查 FIB 表进行转发;
15. end if
```

请求消息传输过程中有两种情况：

a)当请求消息是，初次建立PIT表项时，不做特殊处理，按照NDN路由器传统操作进行。

b)当请求消息聚合PIT表时，为区分不同Interest消息所在路径上的Distance值，需要在PIT表中记录不同的Interest消息的传输跳数(参见表1中第11-12行，请求消息聚合时需要将Interest消息传输跳数记录在PIT表中。当Data消息返回时，如表2中第7行所示，会将Data消息中Distance更新为Data消息传输至此的跳数加上请求消息传输至此的跳数）

表 2 核心区域路由器对 Data 消息的处理算法

核心区域路由器对 Data 消息的处理算法

```
1.  以式（1）计算的概率决定是否缓存当前 Data 消息;
2.  if（需要缓存该内容）
3.     将 Data 消息缓存在 CS 表中;
       CS 表的 Hop = Data 消息 Hop;
4.  end if
5.  for（对于 PIT 每个表项）
6.     将返回的 Data 消息中 Hop+1，复制到新 Data 消息的 Hop 字
       段;
7.     Data 消息 Distance=PIT 表 Hop+Data 消息 Hop;
8.     将数据消息从相应接口转发出去;
9.  end for
```

算法输入和输出

表1算法中的输入是请求消息，输出是数据消息或者是请求消息被转发出去的接口。具体而言，表1算法中的输出分三种情况：a）当请求消息命中缓存时，算法输出是对新生成的数据消息中Hop字段和Distance字段进行修改，并返回新的数据消息（参见表1中3-6行）;b）如果请求消息不命中缓存，且没有匹配的PIT表项，算法输出是创建新的PIT表项（参见表1中第9行），并且查FIB转发出去（参见表1中第14行）;c）如果请求消息不命中缓存，且有匹配的PIT表项，算法输出是将请求消息添加到接口列表，同时重置接口的Hop值（参见表1中第11-12行），并且查FIB转发出去（参见表1中第14行）。

表2算法中的输入是数据消息，输出是数据消息被转发出去的接口。具体而言分两种情况：a）按照式（1）计算当前数据消息的缓存概率，如果决定缓存该数据消息，则存储在CS中，并重置CS表中的Hop字段（参见表2中第3行），并根据PIT表转发数据消息（参见表2中第6-8行）;b）如果决定不缓存该数据消息，则仅需根据PIT表转发数据消息（参见表2中第6-8行）。

命中服务器和命中缓存

用户请求在传输过程中有可能命中服务器，也有可能命中缓存节点。当命中服务器时，*Dcore* 是指从服务器到核心区域最外侧的缓存节点的距离；当命中缓存节点时，此时的 *Dcore* 是指从当前缓存节点到核心区域最外侧的缓存节点的距离。

PIT 聚合问题

当某一请求消息到达某缓存节点，数据还没响应以前，PIT中会有表项记录其到达接口。当再次有请求同一内容的请求消息到达时，PIT 表中会有多条表项，指向不同的到达接口。为计算不同的路径各自的 *Dcore*，在 PIT 表中记录各请求消息到达接口前所经过的跳数 L1。数据消息在返回时记录从内容源返回时的跳数 L2，则 *Dcore* 按照如下方法获得：

$$Dcore= L1+ L2$$

即将 L1 和 L2 相加便可得到 *Dcore*。

3.4 边缘区域的缓存算法

如图 2 所示，在每个边缘区域可以部署一个集中控制节点（可以是 SDN 控制器，但也不是必须是），用于集中管理本区域内的 NDN 路由器。一方面，控制节点负责实时搜集本区域内所有 NDN 路由器上报的内容请求情况、缓存容量、拓扑结构等信息，另一方面控制节点基于搜集信息进行集中计算和决策，并将相应结果告知边缘区域内各 NDN 路由器。

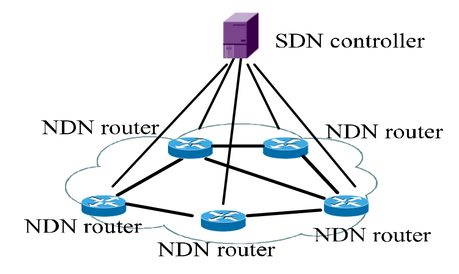


图 2 边缘区域网络架构

具体而言，控制节点及时、快捷地统计到内容热度分布、缓存节点容量、网络拓扑等信息后，基于文献[7]中相应最优化方法，通过控制节点集中计算，可以决策边缘区域内各个缓存节点内具体内容缓存情况，并指导相应 NDN 路由器进行内容缓存。与此同时，为降低计算开销，避免缓存决策频繁变更时对缓存系统产生不利影响，该算法可以阶段性执行，并仅在必要时对路由器内缓存内容进行更新。

算法的输入和输出

算法输入是所有内容的热度排名、各个缓存节点容量、网络拓扑、用户请求情况等信息，算法输出是各个缓存节点应该缓存的内容。

3.5　具体案例

本节对 HDP 方案进行举例说明。如图 2 所示，网络分为核心区域和边缘区域两部分。缓存节点（路由器）v1，v2，v5 在核心区域采用概率性缓存策略，v3，v4，v6 在边缘区域，采用确定性缓存策略。假设所有缓存节点容量均为 10（以内容块为单位）。在最靠近用户的 v4 节点上仅缓存热度排名在 1-10 之间的内容块。在 v3 节点上仅缓存热度排名在 11-20 的内容块。因此，为避免冗余存储，在核心区域 v1 和 v2 仅存热度排名 21 之后的内容。假设此时热度排名为 50 的内容经过 v1 节点时，缓存与否的概率为 25%，经过 v2 节点时，缓存与否的概率为 50%。

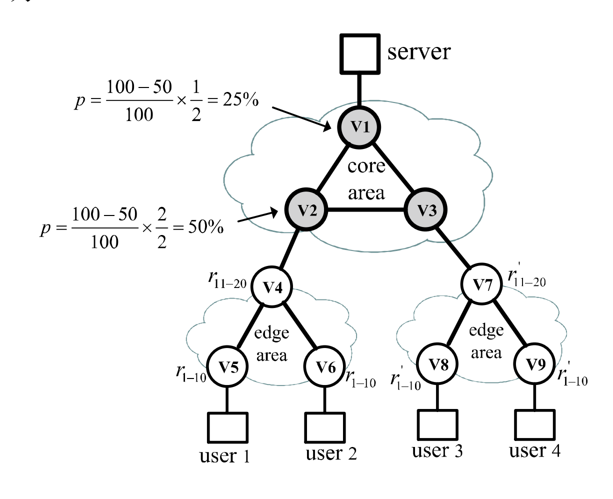


图 3 示例说明

3.6　算法时间复杂度分析

HDP 在网络核心采用概率缓存方法，由于对相关信息的搜集均是线性操作，时间复杂度是 $O(1)$。HDP 在网络边缘采用最优化方法计算缓存策略，时间复杂度是 $O(m^2)$，因此总的时间复杂度不超过 $O(m^2)$，其中 m 为边缘区域内缓存节点数量。LCE、LCD、MCD、RAC 由于缓存操作均是简单的一两步本地操作，时间复杂度为 O(1)。ProbCache 方案在缓存决策时，需要搜集剩余路径容量和当前节点位置信息，时间复杂度为 O(n)，其中 n 为 NDN 网络中缓存节点数目。虽然 HDP 时间复杂度看似比其他方案略高，但考虑到 m 仅代表边缘区域内缓存节点的数量（m<n），因此该时间复杂度并不很高，可以接受。

4　性能评价

4.1　实验环境

开展仿真实验，验证 HDP 的缓存性能。基于 NDN 著名仿真器 NDNSim[15]开展相关实验，将 HDP 与现有确定性和概率性方法进行比较。

如图 4 所示，实验中的网络拓扑结构由通用的 BA(Barabási-Albert)模型构建生成[16]，该拓扑能有效刻画 AS 域内的网络拓扑特征，实验拓扑结构如图 4 所示。假设所有内容块具有相同大小，用 Zipf 分布刻画用户请求（内容热度）情况，表 1 给出了具体参数设置。总共 100 个节点，其中内容源服务器 1 个，用户请求节点（叶子节点）66 个，缓存节点（路由器）33 个。

表 3 参数设置

参数	默认值	变化范围
缓存容量	40	10~100
内容数量	10000	4000~40000
访问模式	Zipf：α=1.0	0.7~1.3
请求速率	200req/s	50~350

假设内容总数为 10000 块，缓存容量默认值为 40，变化范围为 10-100，Zip 参数 α 默认值为 1.0，变化范围为 0.7-1.3，用户请求速率默认值为 200 次/s，,变化范围为 50-350 次/s，度数不超过 3 的节点位于边缘网络，度数不小于 4 的节点位于核心网络。仿真实验执行 60s，将仿真开始 10 秒后的数据作为统计数据。

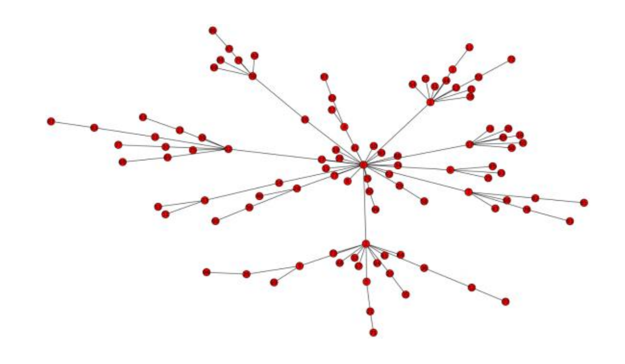


图 4 实验拓扑结构

对比方案有以下几种，其中前三种是确定性缓存方案，后三种是概率性缓存方案：

a）LCE(leave copy everywhere),传统的内容处处缓存方案[1,2]；

b）LCD(leave copy down),内容命中后，向请求方向的下一跳存储相应内容[3]；

c）MCD(move copy down),内容命中后，向请求方向的下一跳存储相应内容，并同时将原内容删除[4]；

d）RAC（0.3）,每个缓存节点以固定概率缓存途经的内容[8]；

e）RAC（0.7）,每个缓存节点以固定概率缓存途经的内容[8]；

f）ProbCache,内容放置的概率与缓存收益以及内容存储能力成正比[9,10]。

不失一般性，本文中所有的缓存方案均采用 LRU 作为缓存替换策略。

4.2 评价指标

为了便于评价HDP缓存方案的性能，在此定义一些相关评价指标。

1）缓存服务率CSR(cache service ratio)

缓存服务率表示被缓存服务的用户请求在所有用户请求中所占的比率。该值越大表明缓存的使用效率越高。

$$CSR=\frac{hit_num}{request_num}\times 100\% \tag{2}$$

其中:hit_num 表示所有用户请求命中缓存的总次数，request_num表示所有用户请求的数量。如果缓存服务率越高，同时也说明缓存系统有助于缓解服务器压力。

2）平均缓存命中率ACHR(average cache hit ratio)

缓存节点n_i的命中率 $CHR_i=\frac{hit_num_i}{request_num_i}\times 100\%$ 。

其中:hit_num_i表示缓存节点n_i被请求命中的次数，$request_num_i$表示缓存节点n_i收到总请求数。

平均缓存命中率 $ACHR=\frac{\sum_{i=1}^{N}CHR_i}{N}$，其中N表示缓存节点的总数。

3）平均访问延迟率AALR(average access latency ratio)

平均访问延迟率表现为有缓存时请求和数据消息的跳数总和与均从服务器取时请求和数据消息的跳数总和的比值，该指标反映系统内容访问的平均响应速率。

$$ARLR=\frac{\sum_{i=1}^{N}hop_s_c_i}{\sum_{i=1}^{N}hop_s_i}\times 100\% \tag{3}$$

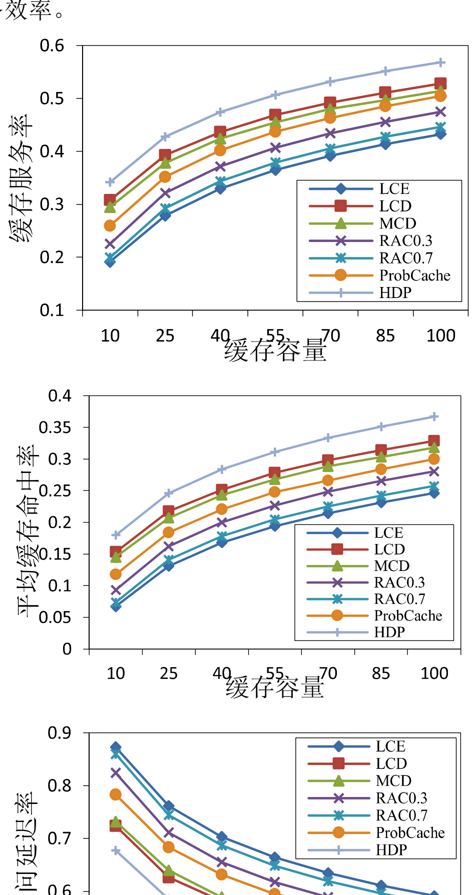
其中:$\sum_{i=1}^{N}hop_s_c_i$ 表示用户从服务器和缓存节点获取内容时，数据消息总的跳数。$\sum_{i=1}^{N}hop_s_i$ 表示所有用户均从服务器获取内容（不从缓存获取）时，数据消息总的跳数。

4.3 实验结果与分析

本节给出相关实验结果，并分别研究缓存容量、内容热度分布、用户请求速率等对缓存性能的影响。为了观察网络性能受某个参数的影响，以下各组实验中我们每次只让单个参数发生变化，其他参数保持不变，参数取值参见表 3。

4.3.1 缓存容量的影响

本节研究缓存容量对各种缓存策略的影响。节点的缓存容量从 10 到 100 个内容块之间变化。从图 5 中可以看出，当缓存容量增长时，所有策略围绕三个指标性能均有所改善。这是由于缓存容量的增长可以更好地发挥缓存的作用。同时 HDP 在各项指标上均好于其他方法。LCD 和 ProbCache 是确定性缓存和概率性缓存中性能最优的。在缓存服务率方面，HDP 相比于 LCD 改善 7.64%-10.9%，HDP 相比于 ProbCache 提高 12.61%-31.59%。平均访问延迟率方面，相比于 LCE 和 ProbCache 分别改善 22.43%-25.14%和 13.5%-16.6%。这是由于 HDP 策略能将概率性缓存和确定性缓存优势相结合，从而能充分提高缓存系统的服务效率。



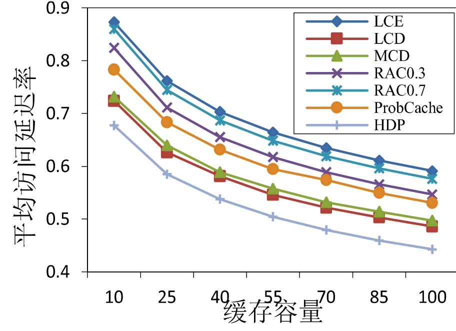


图 5 缓存容量的影响

4.3.2 内容热度的影响

本节研究内容热度（请求模式）对于缓存系统性能的影响。内容热度改变通过 Zipf 分布[17]的参数的调整实现。当 Zipf 参数（α）从 0.7 到 1.3 变化时，实验结果如图 6 所示。

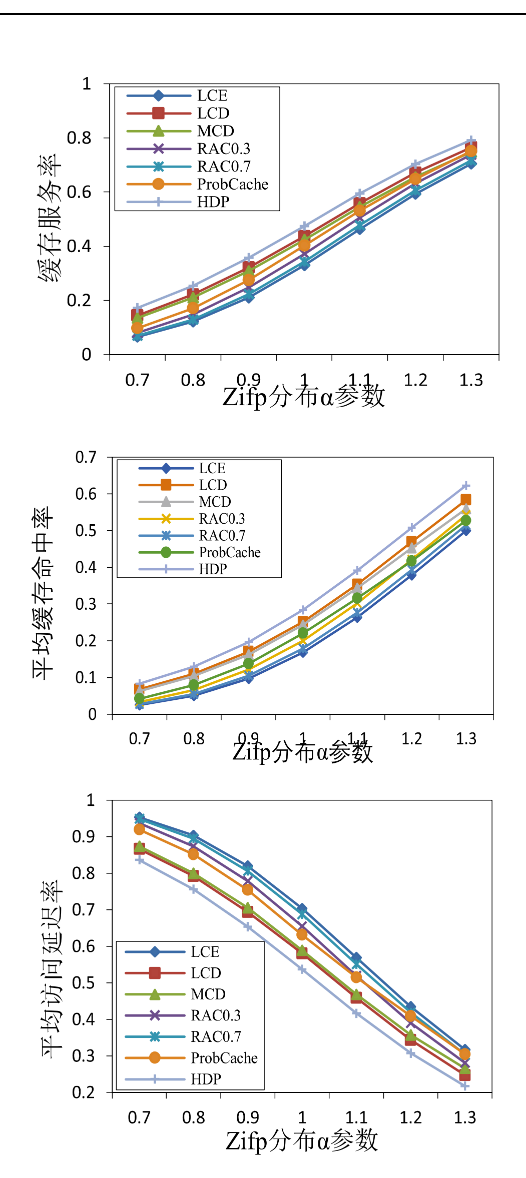


图 6 Zipf 参数（α）的影响

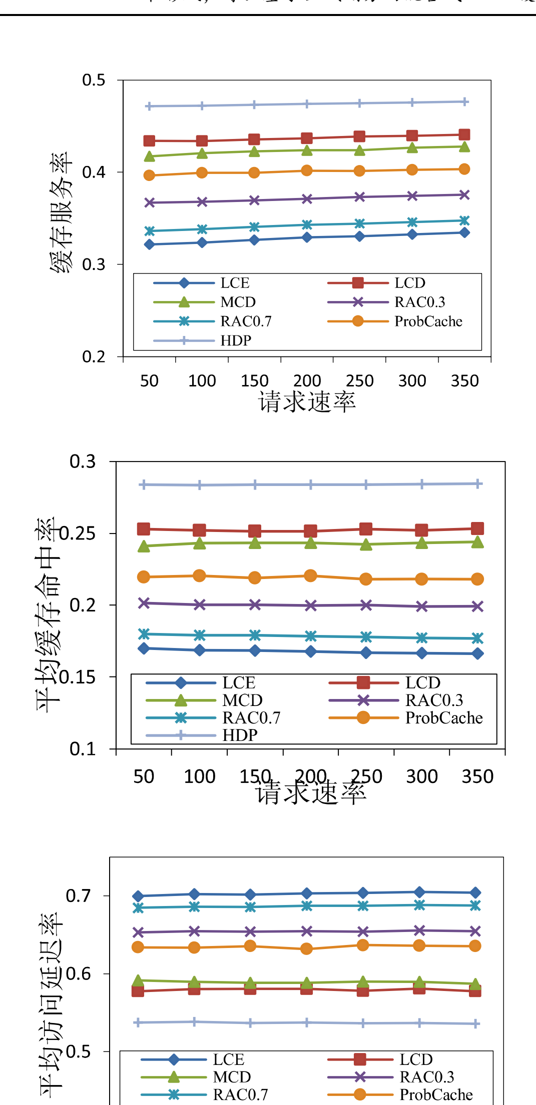


图 7 请求速率的影响

随着 Zipf 参数的增大，各种缓存方法的性能均有所改善，这是由于内容热度差异越大，对内容缓存的效果越显著。三种指标 HDP 策略性能均优于确定性和概率性缓存策略。确定性和概率性概率方法中性能最优的是 LCD 和 ProbCache。HDP 相比于 LCD 和 ProbCache，在缓存服务率方面分别改善 3.5%-19.26%和 5.36%-77.79%，在平均缓存命中率方面分别改善 6.74%-22.74%和 18.05%-97.7%，在平均访问延迟方面分别改善 3.34%-12.13%和 9.04%-28.65%。

4.3.3 请求速率的影响

同时研究了用户请求速率对缓存性能的影响。用户请求速率从 50 个/s 请求到 350 个/s 请求之间变化。如图 7 所示，当请求速率增大时，所有缓存方法在三个考核指标方面均无明显变化，HDP 性能均优于现有确定性和概率性缓存方法。确定性和概率性概率方法中性能最优的是 LCD 和 ProbCache。HDP 相比于 LCD 和 ProbCache，缓存的服务率分别改善约 8.5%和 18.3%，平均缓存命中率分别改善约 12.4%和 30%，平均延迟分别改善 7.4%和 15.5%。

4.3.4 HDP 性能优势分析

LCE、LCD、MCD 等确定性缓存方法，没有对网络的边缘区域和核心区域区分对待，在网络边缘区域基于少量、有限信息进行缓存决策，难以达到较好的缓存效果；同时在网络核心区域容易导致内容重复存储，缓存冗余度过高等问题。

传统 RAC（0.3）、RAC（0.7）等概率方法，由于所有缓存节点机械地采用固定的概率进行缓存，一方面概率值往往仅凭经验，难以取得合理的数值，另一方面固定的概率难以适应网络变化的动态性，因此难以取得理想的缓存性能。概率方法 ProbCache 虽相比于上述两种概率方法有所改进，但由于没有考虑内容热度信息，难以获得较好性能。

本文所提 HDP 方案，采用混合式的缓存策略。一方面，在边缘区域采用确定性缓存方法，借助集中式架构便于获取各种决策信息，并可实现最优缓存方案的快速决策；另一方面，在核心区域采用概率性缓存方法，能有效避免冗余存储，同时便于适应多变的网络环境。同时兼顾内容热度和缓存收益，相比于传统概率方法具有性能改进。上述所分析的性能优势，借助

仿真实验在缓存服务率、缓存命中率、访问延迟等性能指标方面均有有利验证。

5 结束语

针对现有 NDN 网络中确定性缓存和概率缓存方案各自的优缺，提出一种混合式缓存策略，在边缘采用确定性缓存方法，在核心采用概率性缓存方法，通过将两者优势相结合，进一步提高 NDN 网络的缓存性能。实验结果表明，HDP 与现有确定性和概率性缓存方案相比，在缓存服务率、缓存命中率、响应延迟等方面均有显著性能改善。在未来工作中，我们将优化区域划分方法，提出更加精细的划分算法，并计划将 HDP 方案在真实网络环境中进行实验，进一步完善相关实验结果。

参考文献：

[1] Jacobson V, Smetters D K, Thornton J D, *et al.* Networking named content [C]// Proc of the 5th ACM International Conference on Emerging Networking Experiments and Technologies. New York: ACM Press, 2009: 1-12.

[2] Zhang Lixia, Claffy K C, Crowley P, *et al.* Named data networking [J] , ACM SIGCOMM Computer Communication Review, 2014, 44 (3) , 66-73.

[3] Laoutaris N, Syntila S, Stavrakakis I. Meta algorithms for hierarchical web caches//Proc of the 23th IEEE International Performance Computing and Communications Conference. 2004: 445–452.

[4] Laoutaris N, Che Hao, Stavrakakis I. The LCD interconnection of LRU caches and its analysis [J]. Performance Evaluation, 2006, 63 (7): 609-634.

[5] Ming Zhongxing, Xu Mingwei, Wang Dan. Age-based cooperative caching in information-centric networks//Proc of the 31th IEEE International Conference on Computer Communications on Computer Communications Workshop. 2012: 268-273.

[6] Li Zhe, Simon G. Time-shifted TV in content centric networks: the case for cooperative in-network caching//Proc of IEEE International Conference on Communications. 2011: 1-6.

[7] Wang Yonggong, Li Zhenyu, Gareth T, *et al.* Design and evaluation of the optimal cache allocation for content-centric networking [J] , IEEE Trans on Computers, 2016, 65 (1): 95-107.

[8] Arianfar S, Nikander P, Ott J. On content-centric router design and implication//Proc of Re-Architecting the Internet Workshop. 2010: 51–56.

[9] Psaras I, Chai W K, Pavlou G. Probabilistic in-network caching for information-centric networks//Proc of the 2nd Workshop on Information-Centric Networking. 2012: 55-60.

[10] Psaras I, Chai W K, Pavlou G. In-network cache management and resource allocation for information-centric networks [J] , IEEE Trans on Parallel and Distributed Systems, 2014, 25 (11): 2920-2931.

[11] Wang Wei, Sun Yi, Guo Yang, *et al.* CRCache: exploiting the correlation between content popularity and network topology information for ICN caching//Proc of IEEE International Conference on Communications. 2014: 3191-3196.

[12] Lan Wang, Hoque A K M M Cheng, Yi, et al OSPFN: an OSPF based routing protocol for named data networking [R]. University of Memphis and University of Arizona, 2012.

[13] Hoque A K M M, Amin S O, Alyyan A, *et al.* NLSR: named-data link state routing protocol//Proc of ACM Special Interest Group on Data Communication, SIGCOMM Workshop. 2013: 15-20.

[14] Voitalov I, Aldecoa R, Lan Wang, *et al.* Geohyperbolic routing: combining geolocation and hyperbolic information in routing [J]. SIGCOMM Computer Communication Review, 2017, 47 (3): 1-8.

[15] Afanasyev A, Moiseenko l, Zhang Lixia. ndnSIM: NDN simulator for NS-3, NDN, NDN-0005 [R]. 2012.

[16] Lee B, Cao Pei, Fan Li, *et al.* Web caching and Zipf-like distributions: Evidence and implications//Proc of IEEE INFOCOM. 1999: 126-134.

[17] 张国强，李杨，林涛，等. 信息中心网络中的内置缓存技术研究 [J] , 软件学报. 2014, 25 (1): 154−175. (Zhang Guoqiang, Li Yang, Lin Tao, *et al.* Research on the technology of built-in caching in the information centric network [J] Journal of Software. 2014, 25 (1): 154-175.)

[18] Zhang Guoqiang, Wang Xiaohui, Gao Qian, *et al.* A hybrid ICN cache coordination scheme based on role division between cache nodes//Proc of IEEE GLOBECOM. 2015: 1-6.